

LIST XREF Commands

This section covers the following topics:

- General Information
 - Using LIST XREF in Batch Mode
 - Commands for Retrieving Information for Individual Members
 - Command to Report on Program Active References
 - Commands for Verifying the Integrity of an Application
 - Miscellaneous
-

General Information

Functions of the Natural LIST XREF command can be called with commands. There are two ways to enter Natural LIST XREF commands:

- **At the Natural NEXT prompt**
LIST XREF commands must be preceded with 'LIST XREF'. This can be abbreviated to L X (separated by a blank).
- **In the command line of any LIST XREF menu**
LIST XREF commands do **not** have to be introduced with LIST XREF.
Note that any Natural command can be entered after the Command ===> prompt.

In the following sections, the commands are shown as they are to be entered after the Command ===> prompt.

The Basic LIST XREF Command

The basic LIST XREF command - if entered without any command following - initiates the LIST XREF menu dialog.

LIST XREF

Asking for Online Help

Entering LIST XREF ? in response to the NEXT prompt or ? in the Command===> field will cause the XREF command help menu to be displayed:

LIST XREF ?

Notational Conventions

The following rules apply to the syntax descriptions below and in the online help on LIST XREF commands:

- Angle brackets indicate optional parts of the commands; the angle brackets themselves should not be typed. Parentheses may be omitted only when the text they enclose is omitted.
- Keywords shown in upper case can be truncated provided that the input remains unambiguous.
- Text shown in lower case should be replaced by an appropriate value.

In the tables below each syntax description is introduced by the name of the function in the LIST XREF menu.

Using LIST XREF in Batch Mode

All functions of the LIST XREF command can also be executed in batch mode. A batch job has to contain commands in the following order:

- Log on to the library where XREF data is to be retrieved (required), e.g. **LOGON MYLIB**
- Change Natural system parameters (optional), e.g. **GLOBALS PS=60**
- Select a profile (optional), e.g. **L X PROFILE JOHN**
- Select a library structure (optional), e.g. **LX STRUCTURE *R**

Note:

If no profile is selected or the selected profile does not exist, the profile defined as the LIST XREF default profile is used.

- Enter any number of LIST XREF commands, each of them beginning with LIST XREF (or its abbreviation LX), e.g.

L X PROGRAM ACM* USING PROGRAMS VIA CALLNAT
L X VERIFY ALL

Give the FIN command to terminate the Natural session, i.e. **FIN**

Identifying Sets in Batch Mode

If sets are to be processed, a user ID must be specified to identify the set. This can be done in one of the following ways:

- If Natural Security is active, the logon user ID is used to identify the set.
- If Natural Security is not active, an LX PROFILE command can be issued to pass a user ID

Commands for Retrieving Information for Individual Members

Invoked Programs

Function	Code	Syntax
Count references in program	D	COUNT PROG <program> <(program-type)> USING PROG <used-program> <(used-program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property>
Count references to program	C	COUNT PROG <used-program> <(used-program-type)> REF PROG <program> <(program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property>
Program using programs	U	PROG <program> <(program-type)> USING PROG <used-program> <(used-program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property>
Program referenced in programs	R	PROG <used-program> <(used-program-type)> REF PROG <program> <(program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property>
Program using programs recursively	E	PROG <program> <(program-type)> USING REC <used-program> <(used-program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property> <DEPTH depth>
Program referenced in programs recursively	I	PROG <used-program> <(used-program-type)> REF REC <program> <(program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property> <DEPTH depth>
XREF program using programs	X	XREF PROG <program> <(program-type)> USING PROG <used-program> <(used-program-type)> <WITH Subroutine/Entry/Class/Method/Property> <VIA method> <DEPTH depth>
XREF program referenced in programs	Y	XREF PROG <program> <(program-type)> REF PROG <used-program> <WITH Subroutine/Entry/Class/Method/Property> <VIA method> <DEPTH depth>

Methods

The method used to invoke programs. For a list of valid values see table in subsection **Invoked Programs** in the section **Retrieving Information For Individual Members** in this documentation.

Data Areas and Variables

Function	Code	Syntax
Count references to variable	C	COUNT VAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
only for variables defined in global data areas		COUNT GVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
only for variables defined in local data areas		COUNT LVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
only for variables defined in parameter data areas		COUNT PVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
Program using variable	U	PROG <program> <(program-type)> USING VAR <variable> <(structure)> <OF data-area> <USAGE variable-usage>
only for variables defined in global data areas		PROG <program> <(program-type)> USING GVAR <variable> <(structure)> <OF data-area> <USAGE variable-usage>
only for variables defined in local data areas		PROG <program> <(program-type)> USING LVAR <variable> <(structure)> <OF data-area> <USAGE variable-usage>
only for variables defined in parameter data areas		PROG <program> <(program-type)> USING PVAR <variable> <(structure)> <OF data-area> <USAGE variable-usage>
Variable referenced in programs	R	VAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
only for variables defined in global data areas		GVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
only for variables defined in local data areas		LVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
only for variables defined in parameter data areas		PVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
Program using data areas	D	PROG <program> <(program-type)> USING DA-AREA <data-area> <(data-area-type)> <BLOCK data-block>
Data area referenced in programs	E	DA-AREA <data-area> <(data-area-type)> REF PROG <program> <(program-type)> <BLOCK variable-usage>

Views and their Fields

Function	Code	Syntax
Count references to view	C	COUNT VIEW <view> REF PROG <program> <(program-type)> <USAGE ddm-usage>
Count references to field of view	D	COUNT FIELD <field> <(view)> REF PROG <program> <(program-type)> <USAGE field-usage>
Program using views	U	PROG <program> <(program-type)> USING VIEW <view> <USAGE ddm-usage>
View referenced in programs	R	VIEW <view> REF PROG <program> <(program-type)> <USAGE ddm-usage>
Program using fields of views	F	PROG <program> <(program-type)> USING FIELD <field> <(view)> <USAGE field-usage>
Field of view referenced in programs	G	FIELD <field> <(view)> REF PROG <program> <(program-type)> <USAGE field-usage>
Program using data area views	A	PROG <program> <(program-type)> USING DA-VIEW <data-area-view> <(view)> <USAGE data-area-usage>
Data area view referenced in programs	B	DA-VIEW <data-area-view> <(view)> REF PROG <program> <(program-type)> <USAGE data-area-usage>

Copy Code

Function	Code	Syntax
Count references to copy code	C	COUNT COPYCODE <copycode> REF PROG <program> <(program-type)> <USAGE copycode-usage>
Program using copy code	U	PROG <program> <(program-type)> USING COPYCODE <copycode> <USAGE copycode-usage>
Copy code referenced in programs	R	COPYCODE <copycode> REF PROG <program> <(program-type)> <USAGE copycode-usage>

Error Numbers

Function	Code	Syntax
Count references to error number	C	COUNT ERROR <from-error-number> <(error-type)> REF PROG <program> <(program-type)> <THRU to-error-number>
Program using error numbers	U	PROG <program> <(program-type)> USING ERROR <from-error-number> <(error-type)> <THRU to-error-number>
Error number referenced in programs	R	ERROR <from-error-number> <(error-type)> REF PROG <program> <(program-type)> <THRU to-error-number>

Printers

Function	Code	Syntax
Program using printers	U	PROG <program> <(program-type)> USING PRINTER <from-printer-number> <THRU to-printer-number> <NAME printer-name>
Printer referenced in programs	R	PRINTER <from-printer-number> REF PROG <program> <(program-type)> <THRU to-printer-number> <NAME printer-name>

Workfiles

Function	Code	Syntax
Program using workfiles	U	PROG <program> <(program-type)> USING WORK <from-work-file-number> <THRU to-work-file-number> <USAGE work-file-usage>
Workfile referenced in programs	R	WORK <from-work-file-number> REF PROG <program> <(program-type)> <THRU to-work-file-number> <USAGE work-file-usage>

Retained Sets

Function	Code	Syntax
Program using retained sets	U	PROG <program> <(program-type)> USING SET <set> <USAGE set-usage>
Retained set referenced in programs	R	SET <set> REF PROG <program> <(program-type)> <USAGE set-usage>

Processing Rules

Function	Code	Syntax
Count references to rules	C	COUNT RULE <rule> <(rule-status)> REF PROG <map> <(M)>
Map using rules	U	PROG <map> <(M)> USING RULE <rule> <(rule-status)>
Rule referenced in maps	R	RULE <rule> <(rule-status)> REF PROG <map> <(M)>

Natural Resource

Function	Code	Syntax
Count references to Natural resources	C	COUNT NRESOURCE <nresource> REF PROG <program> <type> <USAGE resource-usage>
Program using Natural resources	U	PROG <program> <type> USING NRESOURCE <NRESOURCE> <USAGE resource-usage>
Natural resources referenced in programs	R	NRESOURCE <resource> REF PROG <program> <type> <USAGE resource-usage>

Command to Report on Program Active References

The following command syntax sets the Short list parameter to N and produces an extensive report:

REPORT<program> <(program-type)> <BY user-id> <ON terminal-id>
<FROM yy-mm-dd> <THRU yy-mm-dd >

The following command syntax sets the Short list parameter to Y and produces a condensed report:

SUMMARIZE<program> <(program-type)> <BY user-id> <ON terminal-id>
<FROM yy-mm-dd> <THRU yy-mm-dd>

Commands for Verifying the Integrity of an Application

Verify the Consistency of an Application

Function	Code	Syntax
Programs with inconsistent xref data	X	VERIFY XREF-DATA
Resources used in program	R	VERIFY RESOURCES

Verify the Implementation of an Application

Function	Code	Syntax
Programs not implemented but referenced	P	VERIFY PROGRAM IMPLEMENTATION <(program-type)>
Externals not implemented but referenced	E	VERIFY EXTERNAL IMPLEMENTATION
Sets not built but referenced	S	VERIFY SET BUILT
Error numbers without text	N	VERIFY ERROR TEXT
Programs not impl./ref. starting from one	I	VERIFY PROGRAM STARTING program

Verify the References of an Application

Function	Code	Syntax
Data areas not referenced	D	VERIFY DATA-AREA REFERENCES <(data-area-type)>
Variables in data area not referenced	V	VERIFY VARIABLE REFERENCES <data-area> <(data-area-type)>
Copy code not referenced	C	VERIFY COPYCODE REFERENCES
Error numbers not referenced	N	VERIFY ERROR REFERENCES
Programs not impl./ref. starting from one	I	VERIFY PROGRAM STARTING program
Programs not referenced	P	VERIFY PROGRAM REFERENCES <(program-type)>

Verify the Integrity of an Application

The following command syntax activates all the application verification functions specified in the user's profile:

VERIFY ALL

Miscellaneous

Operate on Sets

Function	Code	Syntax
Start a cataloging process for all objects in the set.	C	CATALOG set-number [KEEP NOKEEP]
Start a stowing process for all objects in the set.	W	STOW set-number [KEEP NOKEEP]
List the source code of all objects in the set.	L	LIST set-number

Option KEEP / NOKEEP

KEEP	The set is retained after processing. The value for parameter Keep set after cat/stow processing from the LIST XREF profile is set to Y for the current session.
NOKEEP	The set is not retained after processing. The value for parameter Keep set after cat/stow processing from the LIST XREF profile is set to N for the current session.

If neither KEEP nor NOKEEP is specified, the current value for Keep set after cat/stow processing is used.

Maintain a User's LIST XREF Profile

PROFILE

Specifying the Library Structure to be used

The following commands are used to specify the library structure which is to be used for evaluating XRef data for subsequent LIST XREF commands. The commands below are only available in batch mode in the XRef menu. Online, the library structure is specified using the parameter Structure as described in the section LIST XREF Main Menu.

- **STRUCTURE <structure-name>**

With this command you can specify which library structure is to be used for evaluating XRef data.

- **STRUCTURE *R**

This command specifies that the runtime structure is to be used for evaluating XRef data.

Note:

If you do not specify any structure, the LIST XREF functions are evaluated without library structure.